

IN THE CLAIMS

1. [Currently amended] An automated software production tool for generating a complete, correct, non ambiguous specification in a formal language which defines a complete operable software program,

comprising:

a software-generating computer programmed to:

receive user input that defines a conceptual model which models a system for which a computer program is to be written;

convert said conceptual model into a high level repository of formal specification statements stated in a formal language having rules of syntax and semantics;

using said computer programmed with a validator program, automatically validating ~~validate~~ said formal language specification using a validator program which, using the rules of syntax and semantics of said formal language, verifies and verifying that every statement in said formal language specification is syntactically complete, semantically correct and not ambiguous.

2. [original] The system according to claim 1, wherein said software-generating computer is programmed to receive said user input using a CASE tool program for presenting a graphical user interface to allow a user to input the requirements of said formal specification using a graphical tools to model said conceptual model graphically.

3. [currently amended] The system according to claim 1 wherein said software-generating computer is further programmed with a system logic translator which controls said computer to process said formal language specification into one or more computer programs that can control a computer to carry out system logic ~~all the~~ functions modeled in said formal specification.

4. [original] The system according to claim 3 wherein said software-generating computer is further programmed with a user interface translator which controls said computer to process said formal language specification into one or more computer programs that can control a computer to implement a particular user interface for the program created by said system logic translator which has been modeled in said formal specification.

5. [original] The system according to claim 3 wherein said software-generating computer is further programmed with a database generator which controls said computer to process said formal language specification into a data structure or database is capable of storing the values of at least all attributes of all objects defined in said formal specification in a manner such that the program or programs generated by said system logic translator can control a computer to read the values of said attributes at any time or store new values for said attributes at any time.

6. [original] The system according to claim 1 wherein said software-generating computer is further programmed by a documentation generator for producing documentation for the software application based on the formal specification.

7. [currently amended] A process for automatic software generation to create a full, operable computer program, comprising:

receiving user input via a computer ~~that defines~~ said user input ~~defining a~~ conceptual model which encodes the requirements ~~the~~ a computer program to be automatically generated must meet and the functions the computer program to be generated must perform;

automatically converting said conceptual model into a high level repository of formal specification statements in a computer file which are stated in a formal language having rules of syntax and semantics; and using said rules of syntax and semantics of whatever formal language in which said formal specification is written to automatically validate said formal specification to ensure that said formal specification is syntactically complete, semantically correct and not ambiguous.

8. [original] The process of claim 7 further comprising the step of using a computer programmed to translate said formal language specification into working system logic computer code that can control a computer to meet the requirements and perform the functions defined in said formal specification.

9. [original] The process of claim 8 further comprising the step of using a computer programmed to translate said formal language specification into computer code that can control a computer to provide a user interface specified in said formal specification.

10. [original] The process of claim 8 further comprising the step of using a computer programmed to translate said formal language specification

into a data structure or data base storing at least the initial values of attributes for objects defined in said formal specification.

11. [currently amended] The process of claim 8 further comprising the step of using a computer programmed to translate said formal language specification ~~specification~~ into one or more requested files of documentation of the program or programs generated by said automatic software generating process.

12. [original] The process of claim 7 wherein the step of receiving user input comprises presenting a graphical user interface having tools by which a user may define the requirements and functions required of the program to be generated using graphical objects displayed on a computer display so as to present

13. [currently amended] A computer readable medium containing instructions for controlling a computer to aid in a process for automatic software generation to automatically write code for a complete, operable computer program, by:

receiving user input defining a conceptual model that defines

~~encodes~~ the requirements and functions of a complete, operable computer program ~~said software~~ to be written;

automatically converting said conceptual model into a formal specification comprised of statements ~~in a computer file~~ written in a formal language having predefined rules of ~~defined~~ syntax and semantics;

using said predefined rules of ~~said defined~~ syntax and semantics to asautomatically validate said formal specification to ensure that it is syntactically complete, semantically correct and not ambiguous; and

~~automatically~~ translating said validated formal specification into one or more computer programs that implement all the system logic, data structures, if any, and user interface functionality, if any, encoded in said formal specification.

14. [currently amended] A computer readable medium containing instructions for controlling a computer to aid in a process for automatic software generation to automatically write code for a complete, operable computer program,, by:

receiving user input defining a conceptual model that defines ~~encodes~~ the requirements and functions of a complete, operable computer program ~~said software~~ to be written;

automatically converting said conceptual model into a formal specification comprised of statements in a computer file written in a formal language having predefined rules of ~~defined~~ syntax and semantics;

using said predefined rules of ~~said defined~~ syntax and semantics to automatically validate said formal specification to ensure that it is syntactically complete, semantically correct and not ambiguous.

15. [currently amended] A method for automatically generating a software application, comprising:

presenting a graphical user interface to allow a user to input requirements for a software application to be automatically generated the ~~software application~~ including user interface patterns;

generating a formal specification in a formal language having predefined rules of syntax and semantics for said the software application based on said the inputs received from said user;

automatically validating said the formal specification to insure it is syntactically ~~syntactically~~ and semantically correct and complete and not ambiguous; and

automatically generating said the software application based on the validated formal specification using one or more translator programs that

translate the validated formal specification into working code in a target computer language.

16. [currently amended] The method according to claim 15, wherein:

said formal specification includes statements defining an application user interface to be presented to a user of the software of the software application; and

the step of generating the software application includes automatic generation of instructions for controlling a computer to present the user interface in accordance with the patterns specified in said the formal specification.

17. [currently amended] A process for forming a functional model that mathematically or logically relates events to the values of attributes the events act upon, comprising the steps:

presenting a dialog box or any other form of user interface that allows a user to specify a class, and an attribute of that class and an event of that class and allows the user to specify mathematical or logical expressions which define a condition and an action, said condition being an expression which when evaluated will result in a true or false logical value, and said action being a

mathematical or logical expression which will generate a new value for the specified attribute in a data type compatible with the data type of said attribute;

receiving user input via said dialog box to specify said class, attribute and event of said class and to specify a single mathematical or logical expression which defines said condition and results in a value which can be mapped to one of only two possible values: true or false, and to specify a single mathematical or logical expression which defines said action and results in a value for said specified attribute in a data type which is compatible with the data type of said specified attribute.

18. [currently amended] A process for forming a functional model that mathematically or logically relates events to the values of attributes the events act upon, comprising the steps:

presenting a dialog box or any other form of user interface that allows a user to specify a class, and an attribute of that class and an event of that class and allows the user to specify mathematical or logical expressions which defines an action, said action being a mathematical or logical expression which will generate a new value for the specified attribute in a data type compatible with the data type of said attribute when said event is executed;

receiving user input via said dialog box or other form of user interface to specify said class, attribute and event of said class and to specify a single mathematical or logical expression which defines said action and results in a value for said specified attribute in a data type which is compatible with the data type of said specified attribute.

19. [currently amended] An apparatus for forming a functional model that mathematically or logically relates events to the values of attributes the events act upon, comprising:

a computer programmed to do the following functions:

present to a user a dialog box or any other form of user interface that allows a user to specify a class, and an attribute of that class and an event of that class and allows the user to specify mathematical or logical expressions which define a condition and an action, said condition being an expression which when evaluated will result in a true or false logical value, and said action being a mathematical or logical expression which will generate a new value for the specified attribute in a

data type compatible with the data type of said attribute; and

receive user input via said dialog box or other form of user interface to specify said class, attribute and event of said class and to specify a single mathematical or logical expression which defines said condition and results in a value which can be mapped to one of only two possible values: true or false, and to specify a single mathematical or logical expression which defines said action and results in a value for said specified attribute in a data type which is compatible with the data type of said specified attribute.

20. [currently amended] An apparatus for forming a functional model that mathematically or logically relates events to the values of attributes the events act upon, comprising:

a computer programmed to do the following functions:

present to a user a dialog box or any other form of user interface that allows a user to specify a class, and an

attribute of that class and an event of that class and allows the user to specify a mathematical or logical which will generate a new value for the specified attribute in a data type compatible with the data type of said attribute when said event is executed; and

receive user input via said dialog box or other form of user interface to specify said class, attribute and event of said class and to specify a single mathematical and/or logical expression which defines how the value of the specified attribute will be changed when the specified event executes.

21. [original] A computer programmed to display a dialog box or any other means by which a user can specify a class, an attribute of that class and a event of that class and to specify a mathematical and/or logical formula which defines how the value of said specified attribute will be changed when said event is executed.

22. [original] The computer of claim 21 wherein said computer is further programmed to

receive user inputs that define the selections of class, attribute and event and said mathematical or logical formula and encode said user inputs into statements in a formal language specification data structure defining the requirements for a computer program to solve a problem to be solved by a computer program.

23. [currently amended] A computer readable medium containing a data structure for storing data that defines the valuation formulas in a functional model which is part of a conceptual model which is used by a computer to automatically generate computer code, said functional model identifying the mathematical or logical relationship between events of classes in said conceptual model and the values of variable attributes said events alter, said data structure comprising any format data which identifies each event which affects the value of a variable attribute and identifies the attribute affected and defines the mathematical and/or logical operations to be performed when said event is executed or occurs and any operands needed for said mathematical or logical operations.

24. [currently amended] An apparatus for creating a graphical user interface to allow user requirements for a computer program to be written by

an automated software production tool to be entered and converted to a formal language specification, comprising:

a software-generating computer programmed to:

display a plurality of dialog boxes and/or graphic screens each of which has boxes which can be filled in with data or menu selections, tools or icons which can be invoked to allow a user to enter information defining classes, attributes, services, relationships between classes, valuation formulas for services that affect the value of variable attributes and all the other information needed to define a conceptual model of the requirements a computer program to be written by said software generation tool must comply with;

receive user input selections and data that defines said conceptual model;

automatically convert said conceptual model into a high level repository of formal specification statements stated in a formal language having predefined rules of syntax and semantics; and

automatically validate said formal specification using